CHAPTER-1
INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 ABOUT ORPHANAGE MANAGEMENT SYSTEM

The title of the project is “Orphanage Management System”. Orphanage management system is an automated and computer-based record system used to capture, process, manipulate, organize orphan data and generate timing, accurate and regular information. This mini project aimed to design and identify the benefit of the development of a computerised orphan record system, a case about many orphanage homes, to capture the orphans and accurate decision. To achieve the goals of the study, data is accomplished by use of ER diagram and Schema diagram.

This management system provides a valuable time saving benefit and do-not have to collect data manually for filling and analysis. Instead of that information can be entered quickly and easily into a computer program and access to the information needed is faster.

The issue of an orphan child is one that bought a lot of worries and concern in recent times. Orphan do-not have siblings or any other forms of relatives, home or identity. For this reason, they lag the usual ambiance enjoyed in the family circle as well as the basic amenities of life and are more or less neglected by the general public.

Inaccuracy since the record are in written form they are bound to be duplication of information, which can resulted to inconsistency as regard to the orphans and vulnerable children record and information.

Orphanage is the name to describe a residential institution devoted to the care of the orphans whose parents are deceased. Parent and sometimes grandparents are legally responsible for supporting children, but in the absence of their or other relatives willing to care for the children, they lose all protection. Orphanage provides an alternative to foster care or adoption by giving orphans a community-based setting in which they live and learn. This project proposed an orphanage management information system that will change the conventional manual management to a computerized management system. This will facilitate inform acquisition, storage and retrieve using Java programming and SQL languages is used for user interface design and it provides the necessary codes in achieving the aims of the system.

1.2 About SQL

- Structured Query Language (SQL) is comprehensive database language. Hence it has both DDL and DML.
- Data Definition Language (DDL): We can use CREATE, INSERT, DELETE MODIFY statements. We cannot manipulate the data in the table.
- Data Manipulation Language (DML): We can manipulate the data in the record using UPDATE and ALTER statements.
- SQL has several different techniques for writing programs in various Programming languages that include SQL statements to access one or more database.
- SQL has transaction control commands. These are used to specify units of database processing for concurrency control and recovery purpose.

MySQL is an relational database management system(RDBMS).Its name is a combination of "My", the name of co-founder Michael Widenius's daughter and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larsson and Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from mSQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as mSQL

MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer.

Mysql dump is a logical backup tool included with both community and enterprise editions of MySQL. It supports backing up from all storage engines. MySQL Enterprise Backup is a

hot backup utility included as part of the MySQL Enterprise subscription from Oracle, offering native InnoDB hot backup, as well as backup for other storage engines.

Xtra Backup is an open-source MySQL hot backup software program. Features include hot, non-locking backups for InnoDB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of I/O operations per second, etc.

MySQL Fabric is an integrated system for managing a collection of MySQL servers, and a framework on top of which high availability and database sharding is built. MySQL Fabric is open-source, and supports procedure execution in the presence of failure, providing an execution model usually called *resilient execution.*

Working with the JAVA Database

This document demonstrates how to set up a connection to java DB in Netbeans IDE. Once a connection is made ,you can begin working with the database in the IDE, allowing you to create tables, populate them with data , run SQL statements and queries, and more. The Java DB is sun's supported distribution of **Apache Derby**. Java DB is a fully transactional, secure, standards-based database server, written entirely in java ,and fully supports SQL ,JDBC API, and java EE technology.The Java DB is packaged with the Glass Fish application server ,and is included in JDK 6 as well.

1.3 Scope and Applications of Orphanage Management System.

This Database plays an important role in Managing the orphans that can greatly improve the accuracy, security and efficiency of the Information Management of Orphans. Mini projects can be developed using JAVA, PYTHON, PHP, HTML, CSS etc., but mainly we are discussing here about Java.

Orphanage management system is an automated and computer-based record system used to capture, process, manipulate, organize orphan data and generate timing, accurate and regular information.
This mini project aimed to design and identify the benefit of the development of a computerised orphan record system, a case about many orphanage home, to capture the orphans and accurate decision. To achieve the goals of the study. Data is accomplished by us of ER diagram and Schema diagram.

This management system provides a valuable time saving benefit and do-not have to collect data manually for filling and analysis. Instead of that information can be entered quickly and easily into a computer program and access to the information needed is faster. The issue of an orphan child is one that bought a lot of worries and concern in recent times. Orphan do-not have siblings or any other forms of relatives, home or identity. For this reason, they lag the usual ambiance enjoyed in the family circle as well as the basic amenities of life and are neglected by the general public. Inaccuracy since the record are in written form, they are bound to be duplication of information, which can result to inconsistency as regard to the orphans and vulnerable children record and information.

CHAPTER-2
LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

- Lack of immediate retrievals: In the conventional system, information is distributed across several files. This might also lead to data redundancy with repetition of the same information in various files. In the event of a complex or nested query, the search has to scan several files, thus making procurement of requested query results very cumbersome.

- Maintenance of Accuracy and Reliability issues: With redundancy comes consistency issues as the update of information in a single record should be echoed in all records containing the same information. Also, atomicity issues i.e., completion of a transaction in totality or nothing at all; has to be maintained. This is difficult in a multi-file system.

- Lack of prompt update: Updates associated with a record in a file is to be reflected in all records wherein the record is present. This concurrent update poses the problem of time lag. Errors in commit operation to some files cause the grave issue of data inconsistency.

- Error prone manual calculation: Manual calculations are error prone and relatively immensely time consuming, in spite of which they may result in generation of incorrect information.

Verification is another overhead, which can be saved through efficient design and implementation.

Improved manual System: One of the alternative solutions is the improvement of the manual system. Anything, which can be done by using automated methods, can be done manually. But the key question is how to perform a task manually in a sound and optimal manner. Following are some suggestions, which can be useful in manual system.

A more sophisticated register maintenance dedicated to each subsystem can be maintained with centralized control and evolution. Adequate dedicated staff may be maintained so that updates are made at very moment at the same time. Proper provision for proper work should be put into place.

This would require considerable extra work force.

Batch system: An alternative solution can be used of computer-based batch system for maintaining the information regarding personal details. A batch system refers to a system in which data is processed in a periodical basis. The batch system can achieve most of the goals and sub goals. But a batch system data is processed in sequential basis. Therefore, batch system is not suggested.

Online system: This system (HRDBMS) provides online storage, update and retrieval facility. This system promises very less or no paperwork and also provides help to administrate and company personal. In the system everything is stored electronically so constants of paperwork are eliminated and multiple files usage is not required. Information can be retrieved very easily without scanning multiple registers this system is been discussed here.

2.2 PROBLEM STATEMENT:

Usually in present system or existing system the admin of the orphanage homes have to maintain all records of orphans manually and there is chances of losing all this data by one or the other reason and for every information a volunteer, staff, donator, adopter have to visit the orphan home and get the data or updates from the record book and volunteer or donor cannot involve in the system when they are out of station. There is no accuracy, consistency and efficiency in the present management system. Present system is time consuming and need more storage space and there is a chance of duplication of the records and anyone can easily modify or chance the data since it is in paper pen format.

2.3 PROPOSED SYSTEM:

In this new Orphanage Management System much of manual processing repetition work has been eliminated data redundancy, save time and storage area. Here the orphan's details can easily be updated or deleted by the authorized user. The admin can easily add

the orphans where each orphan will be having unique orphan_id. The updating and deletion operation are so easy. If the authorized user needs any information, he/she can easily search the information where the search portal is provided for each table. We can Generate a receipt of each orphan and can also be resettled.

2.4 ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Authorized user enter his/her user ID and password for login to authenticate in this system
- This system will replace the traditional file system.
- Authorized user can have his own home page regarding the information about orphan, staff, donors, volunteers, adopters, requirements.
- Authorized user can create the perfect various orphan details in this website.
- Authorized user can create various details like staff, donors, adopters, requirements in this website.
- Admin can enter every personal detail of orphan.
- Authorized user gets all orphan and other details.
- The registered users can be stored in centralized database through the system user interface.
- Various types of information can be displayed to the users.
- The authorized user can search for the orphan information, staff information and many other.
- Searching technique which is one of the most unique technique of this project which provides the admin to fetch the orphan details easily.
- Only admin have all the authority to add, update, search, delete and view the details.
- Less utilization of space.

DISADVANTAGES

- In this System everything is controlled by admin so there will be a struck in the process in the absence of admin.
- Only skilled people can use this system.
- Everyone cannot add or modify the data of the orphanage home.
- Any missing data is not accepted.
- Many rules and regulations have to be followed.

CHAPTER-3
HARDWARE, SOFTWARE AND FUNCTIONAL REQUIREMENTS

CHAPTER 3

HARDWARE, SOFTWARE AND FUNCTIONAL REQUIREMENTS

The program works on Desktop PC and is executed using a java swing interface which interacts with a MYSQL database running on local host.

3.1 HARDWARE REQUIREMENTS:

CPU : Pentium IV 2.4 GHz or above

Memory : 512 MB,1GB or above

Hard Disk : 40GB, 80GB,160GB or above

Monitor : 15 VGA color

3.2 SOFTWARE REQUIREMENTS:

Programming language : JAVA, MYSQL

Operating system : Any OS (Recommended: windows 8,Windows Vista,windows XP

Application required : Standalone desktop application and Xampp

Coding language : core JAVA

3.3 FUNCTIONAL REQUIREMENTS

A description of the facility or feature required. Functional requirements deal with what the system should do or provide for users. They include description of the required functions, outlines of asso

3.3.1 Interface Requirements:

- The system shall provide an option to add/delete items.
- The system should give option for login.
- The system shall provide option to add/delete suppliers.
- The system shall provide option to automatically restock the inventory.

3.4 NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements define the overall qualities or attributes of the resulting system.

3.4.1 Usability

Usability is the ease with which a user can learn to operate the warehouse inventory management system and get results.

3.4.2 Security

Security requirements are included in a system to ensure:

- All inventory and supplier information are well secured
- SQL injection is prevented

3.4.3 Reliability

Reliability is the ability of a system to perform its required functions under stated conditions for a specific period of time. Constraints on the run-time behavior of the system can be considered under two separate headings:

- Availability: is the system available for service when requested by end-users.
- Failure rate: how often does the system fail to deliver the service as expected by endusers.

3.4.4. Efficiency

The comparison of what is actually produced or performed with what can be achieved with the same consumption of Clouds (money, time, labor, etc.). It is an important Factor in Determination of Productivity.

CHAPTER-4
ANALYSIS AND DESIGN

CHAPTER 4

ANALYSIS AND DESIGN

4.1 ARCHITECTURE OF PROPOSED SYSTEM

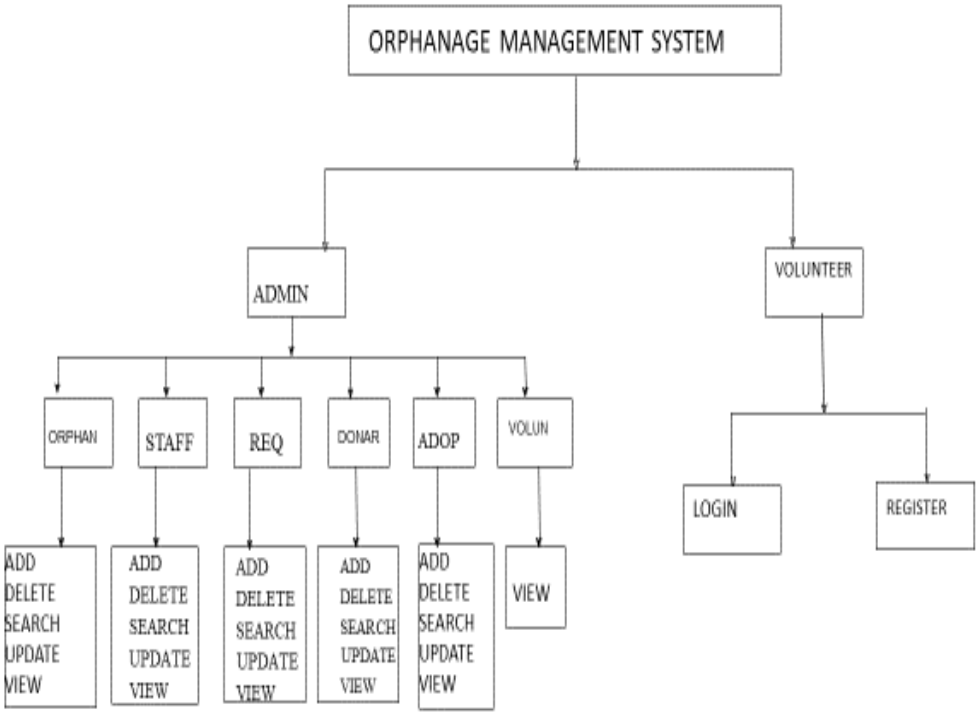

FIG 4.1: ARCHITECTURE OF ORPHANAGE MANAGEMENT SYSTEM.

SYSTEM DESIGN

System Design is the most creative and challenging phase in the system life cycle. Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. System design is a solution how to approach the creation of a new system. System design transforms a logic representation of what is required to do into the physical specification. The specification is converted into physical reality during development.

Logical design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volume's, frequencies etc.
- Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

Physical design

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design Walk

through.

- Plan system implementation.
- Prepare a conversion schedule and target date.
- Determine training procedures, courses and timetable.
- Devise a test and implementation plan and specify any new hardware/software.

Input design

- Authorized User enter his/her user id and password for login to authenticate in this system
- User Enters the Orphan Details, While registering information of the orphan such as
- Orphan id
- Orphan name
- Orphan age
- Orphan gender etc...
- Staff details such as name, age, doj, staff id etc,,,
- Requirement details name, expense, quantity, etc…
- Donor details such as name, donating date, etc…
- Adopter details such as name, id, address, adopting date etc, …
- Volunteer details can be viewed and can view volunteering.
- And volunteer can register and can involve in volunteering activity.

Output design

- Authorized User can have his own home page regarding the information about the orphan, staff adopter, donor, requirement and volunteer.
- Authorized User get all details.
- The registered user's data can be stored in centralized database through the system user interface.

- Various types of information can be displayed to the users like orphan information, staff information, donor information, adopter information, volunteer information.
- The Authorized User can search for the orphan, staff, donor, adopter ,volunteer information.
- Profile can be updated by the users individually.
- Searching technique which is one of the most unique technique of this project provide the user to easily fetch the information of all the entities.
- Authorized User can also add another user.
- The Authorized User has all the authority to update, delete the information of the orphan, staff, donor, volunteer.

4.2 ENTITY-RELATIONSHIP DIAGRAM AND SCHEMA DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

There are two reasons to create a database diagram. You're either designing a new schema or you need to document your existing structure.

If you have an existing database you need to to document, you create a database diagram using data directly from your database. You can export your database structure as a CSV file (there are some scripts on how to this here), then have a program generate the ERD automatically.

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.
- A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.
- **Actions**, which are represented by diamond shapes, show how two entities share

information in the database.

- In some cases, entities can be self-linked. For example, employees can supervise other employees.

- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.

- A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.
- A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.
- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
- **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality.

ENTITY RELATIONSHIP DIAGRAM

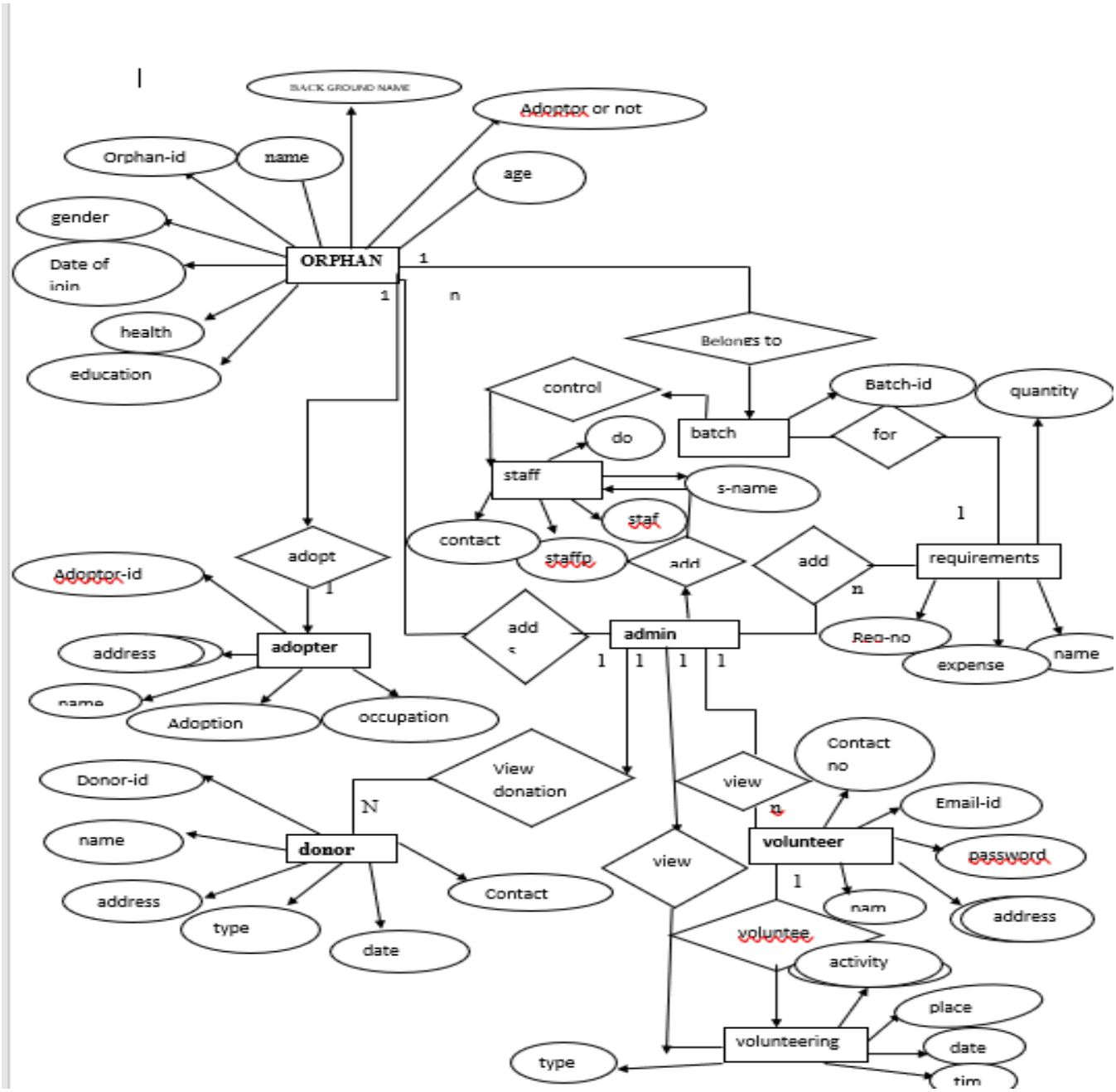

FIG 4.2.1: ENTITY-RELATIONSHIP DIAGRAM

SCHEMA DIAGRAM

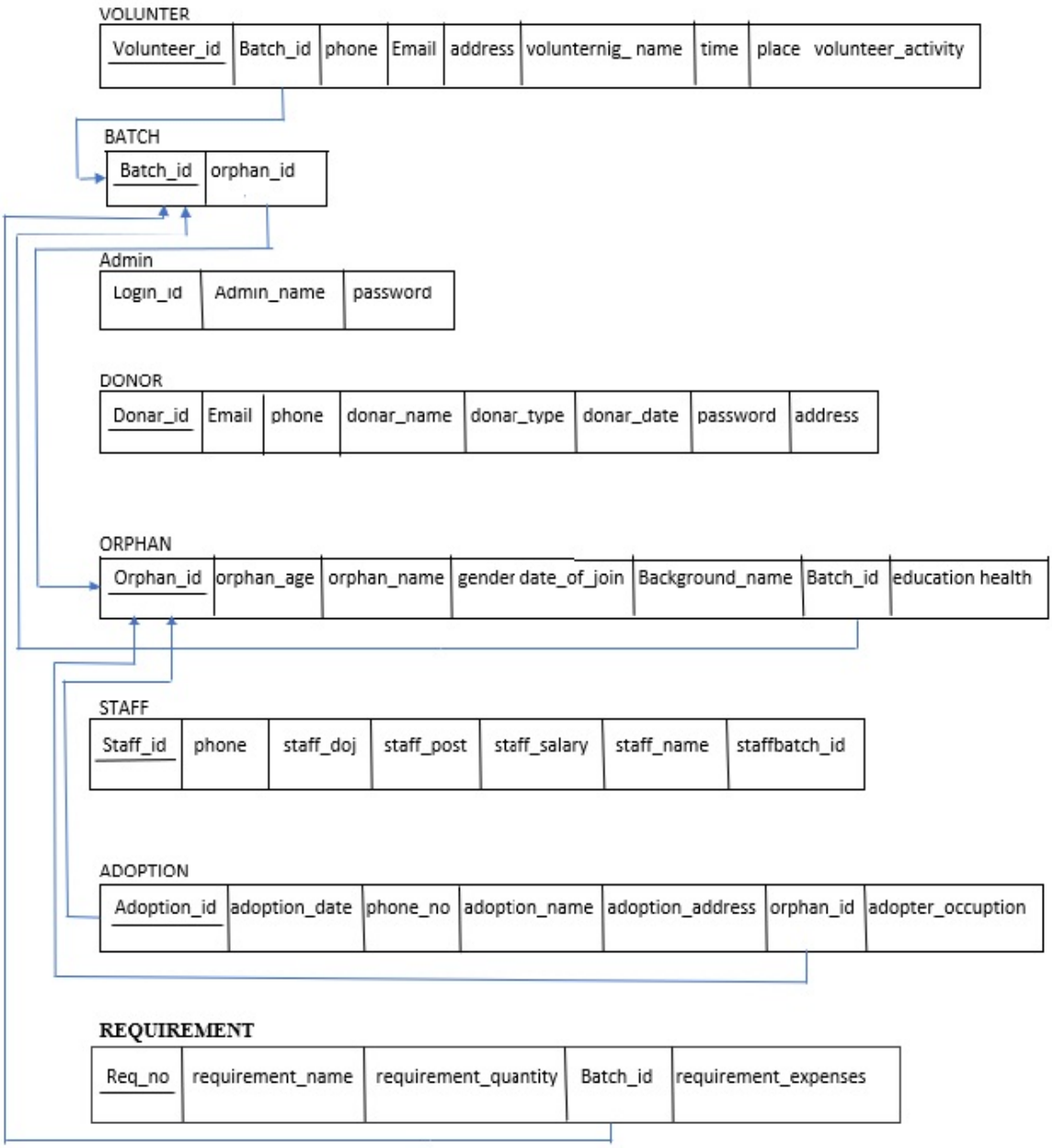

FIG 4.2.2: SCHEMA DIAGRAM

CHAPTER-5
IMPLEMENTATION

CHAPTER 5

IMPLEMENTATION

5.1 MODULES DIVISION

This project is modularized as the following:

1. orphan Entity`
2. donor entity
3. adopter entity
4. volunteer entity
5. requirement entity
6. staff entity.
7. Admin

1. **orphan entity:** The orphan Record Storage and Retrieved by this Module. This Module include orphan record Information Insert, View, Delete, Update By User friendly Approach.
2. **donor entity:** The donor Entry and View donor Detail by this Module. This includes donor Information Storage and maintaining their information. Form validation is already Done by this project.
3. **staff entity:** The common and fixed information related to Department. We can categories each staff based on post. Then Attached to Department to identify or make Search easier.
4. **volunteer entity:** The volunteer entity contains details of volunteering activities, and manage registration, view volunteering activities.
5. **Adopter entity:** By this Module we can insert adopter details and the orphan details.
6. **Requirement entity:** By this module we will get to know what requirements needed in what quantity.
7. **Admin:** Admin entity controls other entities, maintains whole database.

5.2 ALGORITHM OR PSEUDOCODE OF ABOVE MODULE

ADMIN ENTRY:

This program will allow the admin to enter the username and password.

If the entered credentials are correct, then the login will be successful otherwise unsuccessful.

After successful login the admin will be redirected to Admin portal page where he can do the following activities.

ORPHAN ENTRY:

This program will allow the user to enter the orphan details.

```
 If id, name, age, doj, education, health, gender, batch_id,background,adop_or_not is entered
    Insert the values into orphandetails
        Else print error
While update=true
         Update the values that has been entered by altering the orphan_details
              Else print update error
View the details of orphan that has been entered by clicking onto view orphan button.
If we need to delete the particulars that has been entered click onto delete button.
If we need to search the particulars that has been entered click onto search button.
If we need to update the particulars that has been entered click onto update button.
```

STAFF ENTRY:

This program will allow the user to enter the staff details.

```
 If id, name, post, phoneno, salary, batch_id, doj is entered
    Insert the values into staff
        Else print error
While update=true
         Update the values that has been entered by altering the staff
              Else print update error
View the details of staff that has been entered by clicking onto view staff button.
If we need to delete the particulars that has been entered click onto delete button.
If we need to search the particulars that has been entered click onto search button.
```

If we need to update the particulars that has been entered click onto update button.

DONOR ENTRY:

This program will allow the user to enter the donor details.

If id, name, phone, address, donartype, donating date is entered

Insert the values into donor

Else print error

While update=true

Update the values that has been entered by altering the donor

Else print update error

View the details of donor that has been entered by clicking onto view donor button.

If we need to viewdonations the particulars that has been entered click onto viewdonations button.

If we need to search the particulars that has been entered click onto search button.

If we need to update the particulars that has been entered click onto update button.

REQUIREMENT ENTRY:

This program will allow the user to enter the requirement details.

If id, name, quantity, expense, batchno is entered

Insert the values into requirement

Else print error

While update=true

Update the values that has been entered by altering the requirement

Else print update error

View the details of requirement that has been entered by clicking onto view requirement button.

If we need to delete the particulars that has been entered click onto delete button.

If we need to search the particulars that has been entered click onto search button.

If we need to update the particulars that has been entered click onto update button.

ADOPTER ENTRY

This program will allow the user to enter the adopter

If id, name, orphaned, address, phone, adoptiondate, adopteroccupation is entered

Insert the values into adopter

Else print error

While update=true

Update the values that has been entered by altering the adopter

Else print update error

View the details of adopter that has been entered by clicking onto view adopter button.

If we need to delete the particulars that has been entered click onto delete button.

If we need to search the particulars that has been entered click onto search button.

If we need to update the particulars that has been entered click onto update button.

VOLUNTEER VIEW:

This program will allow the user to view volunteer details and view volunteering.

VOLUNTEER REGISTER:

In this program the volunteer will register for the orphanage management system to conduct volunteer activities.

VOLUNTEER LOGIN:

In this program the volunteer enters the volunteering details like volunteer type, activity, date, timings, place.

CHAPTER-6
DISCUSSION AND SNAPSHOTS

CHAPTER 6

DISCUSSION AND SNAPSHOTS

Tables:

The given below table is a snapshot of backend view of localhost and the structures of the tables present in Orphanage Management System. The tables present are admin, batch, orphandetails, staff, requirements, adopter, donor, volunteer, volunteerr, staff_audit.

The admin is used to store the information about the admin.

The orphandetails is used to store the information about the orphan.

The staff is used to store the information about the staff.

The batch is used is store the information about the batches.

The requirements is used to store the information about the requirements required by the orphan home.

The adopter is used to store the information about the adopters.

The donor is used to store the information about the donors.

The volunteer is used to store the information about the volunteers.

The volunteerr is used to store the information about the volunteering activities.

```
mysql> show tables;
+------------------+
| Tables_in_orphan |
+------------------+
| admin            |
| adopter          |
| batch            |
| donor            |
| orphandetails    |
| requirements     |
| staff            |
| staff_audit      |
| volunteer        |
| volunteerr       |
+------------------+
10 rows in set (0.04 sec)
```

FIG 6.1: List of tables

Admin Table:

```
mysql> desc admin;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(20) | NO   | PRI | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.04 sec)
```

FIG 6.2: ADMIN TABLE DESCRIPTION

Orphandetails:

```
mysql> desc orphandetails;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| orphan_id       | varchar(20) | NO   | PRI | NULL    |       |
| orphan_name     | varchar(20) | YES  |     | NULL    |       |
| orphan_age      | int(11)     | YES  |     | NULL    |       |
| gender          | varchar(20) | YES  |     | NULL    |       |
| date_of_join    | varchar(10) | YES  |     | NULL    |       |
| health          | varchar(20) | YES  |     | NULL    |       |
| education       | varchar(20) | YES  |     | NULL    |       |
| background_name | varchar(20) | YES  |     | NULL    |       |
| batch_id        | varchar(20) | YES  |     | NULL    |       |
| adopt_or_not    | char(20)    | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

FIG 6.3: ORPHAN TABLE DESCRIPTION

Staff:

```
mysql> desc staff;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| staff_id    | varchar(20) | NO   | PRI | NULL    |       |
| staff_name  | varchar(20) | YES  |     | NULL    |       |
| staff_phone | varchar(20) | YES  |     | NULL    |       |
| staff_post  | varchar(20) | YES  |     | NULL    |       |
| staff_doj   | varchar(20) | YES  |     | NULL    |       |
| onbatch     | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

FIG 6.4: ORPHAN TABLE DESCRIPTION

Requirements:

```
mysql> desc requirements;
+----------------------+-------------+------+-----+---------+-------+
| Field                | Type        | Null | Key | Default | Extra |
+----------------------+-------------+------+-----+---------+-------+
| req_no               | varchar(20) | NO   | PRI | NULL    |       |
| requirement_name     | varchar(20) | YES  |     | NULL    |       |
| requirement_quantity | varchar(20) | YES  |     | NULL    |       |
| requirement_expense  | int(11)     | YES  |     | NULL    |       |
| onbatch              | varchar(20) | YES  |     | NULL    |       |
+----------------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

FIG 6.5: REQUIREMENT TABLE DESCRIPTION

Adopter:

```
mysql> desc adopter;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| adopter_id         | varchar(20) | NO   | PRI | NULL    |       |
| adopter_name       | char(20)    | YES  |     | NULL    |       |
| orphan_id          | varchar(20) | YES  |     | NULL    |       |
| adopter_address    | varchar(40) | YES  |     | NULL    |       |
| phoneno            | varchar(10) | YES  |     | NULL    |       |
| adopt_date         | varchar(10) | YES  |     | NULL    |       |
| adopter_occupation | char(20)    | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

FIG 6.6: ADOPTER TABLE DESCRIPTION

Donor:

```
mysql> desc donor;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| donor_id      | varchar(20) | NO   | PRI | NULL    |       |
| donor_name    | char(20)    | YES  |     | NULL    |       |
| donor_phone   | varchar(20) | YES  |     | NULL    |       |
| donor_address | varchar(50) | YES  |     | NULL    |       |
| donating_type | varchar(20) | YES  |     | NULL    |       |
| donating_date | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

FIG 6.7: DONOR TABLE DESCRIPTION

Batch:

```
mysql> desc batch;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| batch_id | varchar(20) | NO   | PRI | NULL    |       |
+----------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

FIG 6.8: BATCH TABLE DESCRIPTION

Volunteerr:

```
mysql> desc volunteerr;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| email_id   | varchar(20) | NO   | PRI | NULL    |       |
| name       | char(20)    | YES  |     | NULL    |       |
| contact_no | varchar(20) | YES  |     | NULL    |       |
| address    | varchar(50) | YES  |     | NULL    |       |
| password   | varchar(10) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

FIG 6.9: VOLUNTEER TABLE DESCRIPTION

Volunteer:

```
mysql> desc volunteer;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| email_id           | varchar(20) | YES  |     | NULL    |       |
| volunteer_type     | varchar(20) | YES  |     | NULL    |       |
| volunteer_activity | varchar(20) | YES  |     | NULL    |       |
| date               | varchar(20) | YES  |     | NULL    |       |
| timings            | varchar(20) | YES  |     | NULL    |       |
| place              | varchar(20) | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

FIG 6.10: VOLUNTEERING TABLE DESCRIPTION

Staff_audit:

```
mysql> desc staff_audit;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | int(11)     | NO   | PRI | NULL    | auto_increment |
| staffnumber | varchar(20) | NO   |     | NULL    |                |
| firstname   | char(50)    | NO   |     | NULL    |                |
| changedat   | datetime    | YES  |     | NULL    |                |
| action      | varchar(50) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
5 rows in set (0.03 sec)
```

FIG 6.11: STAFF AUDIT(TRIGGER) TABLE DESCRIPTION

SNAPSHOTS:

FIG 6.12: HOME PAGE OF ORPHANAGE MANAGEMENT SYSTEM.

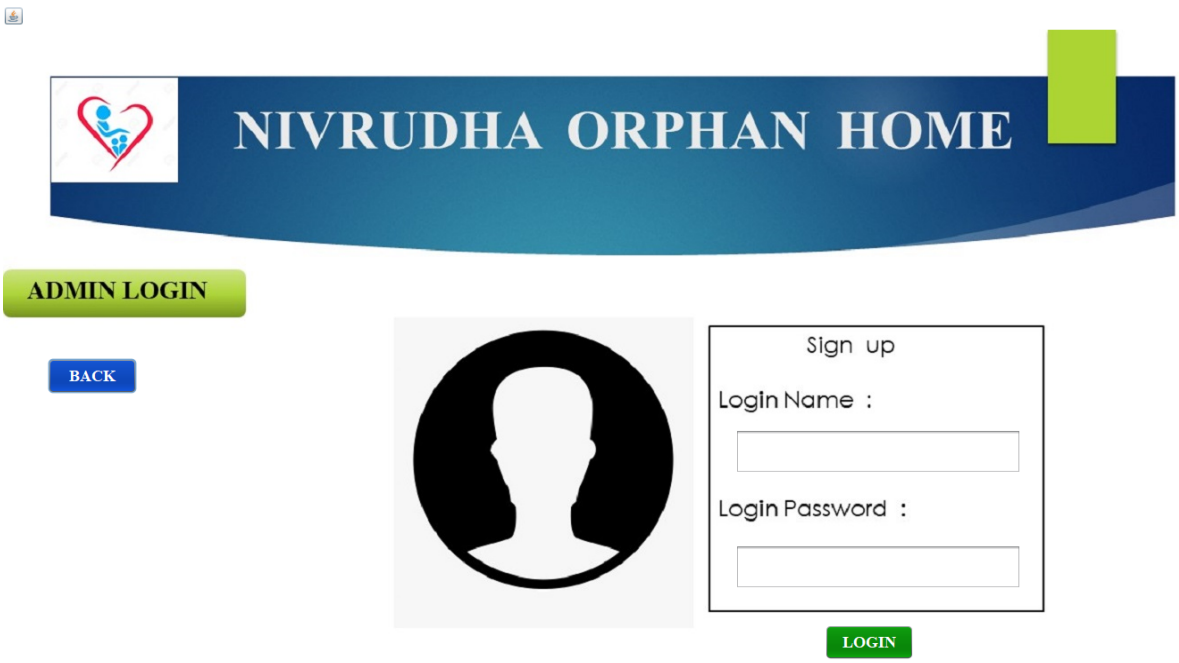

FIG 6.13: ADMIN LOGIN PAGE.

FIG 6.14: ADMIN PORTAL PAGE.

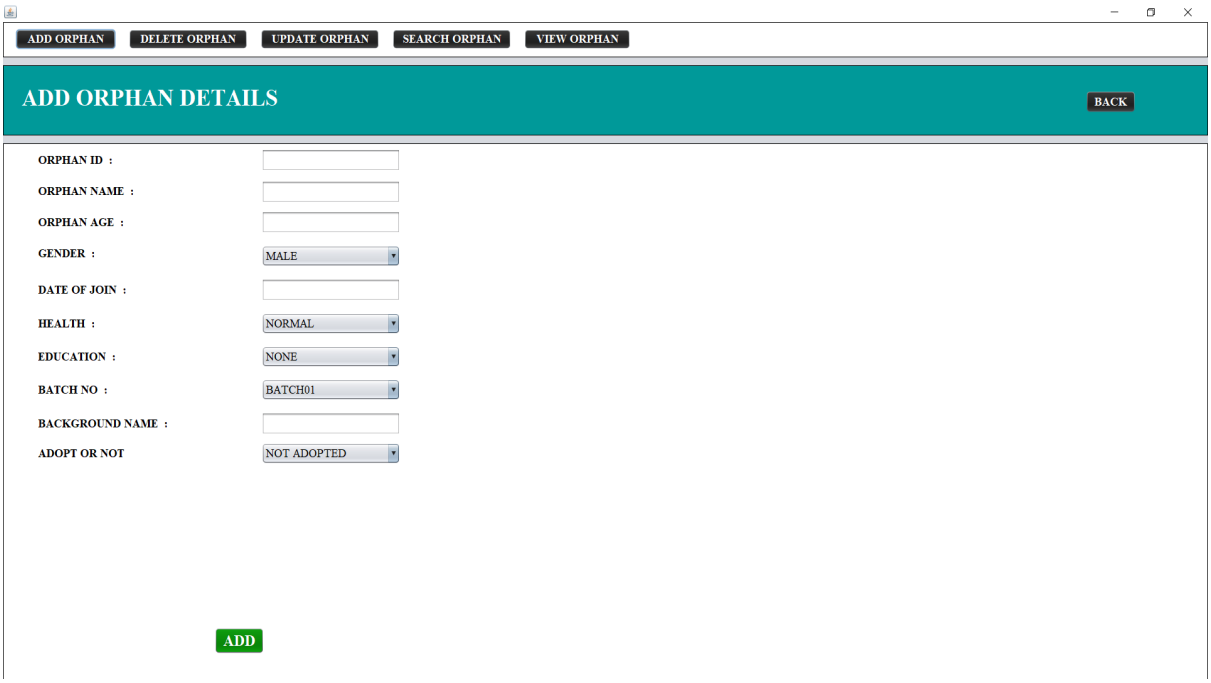

FIG 6.15: ORPHAN DETAILS

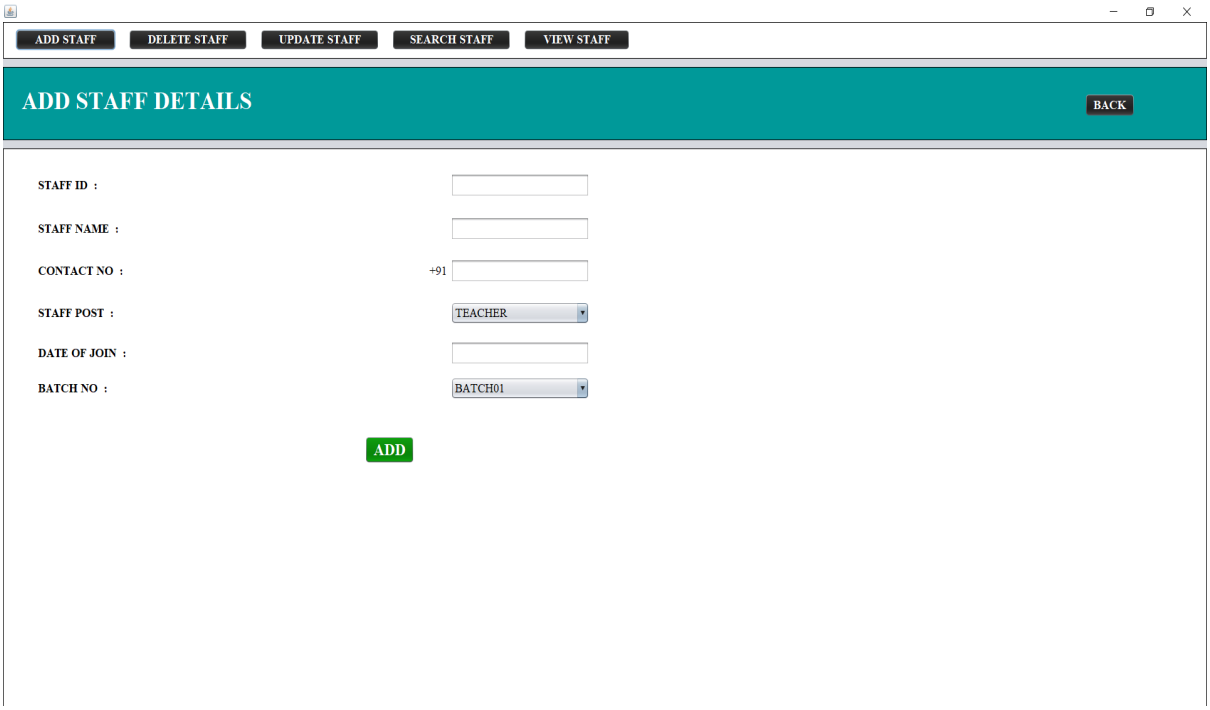

FIG 6.16: STAFF DETAILS

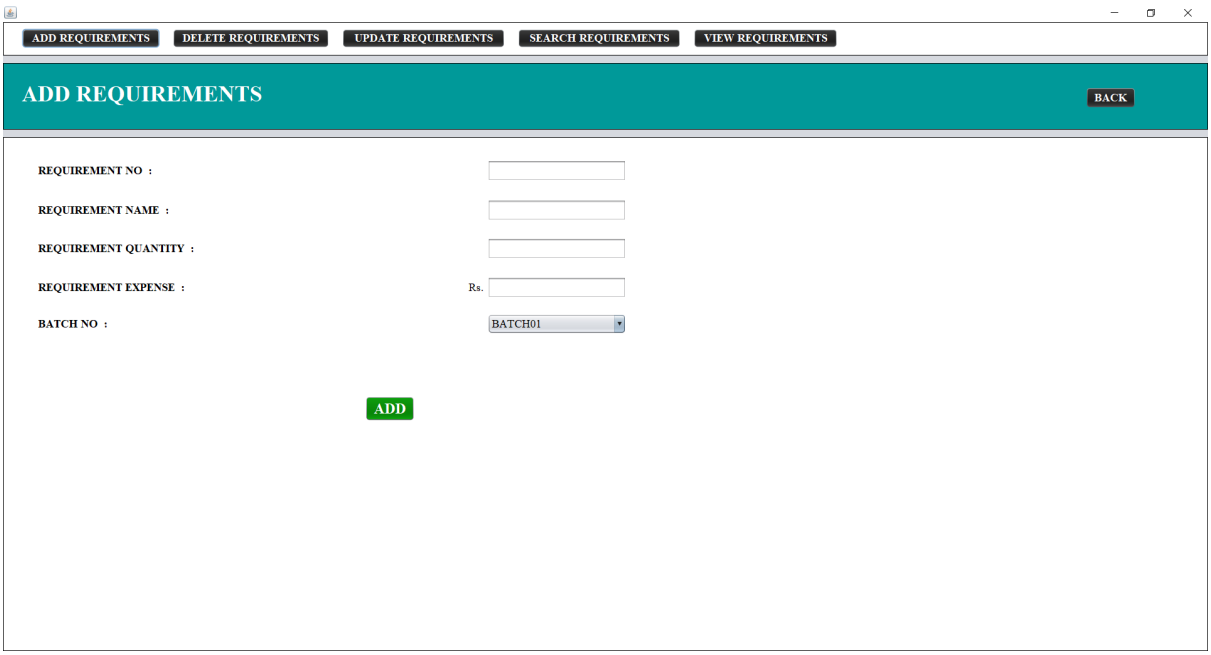

FIG 6.17: REQUIREMENTS DETAILS

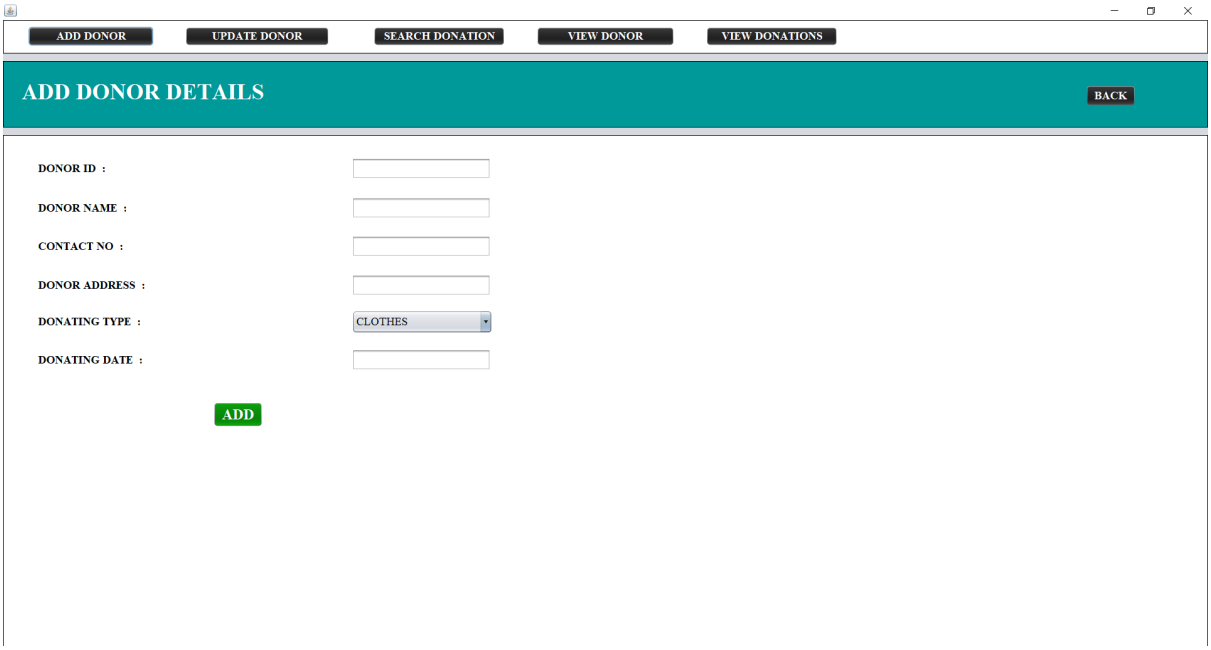

FIG 6.18: DONOR DETAILS

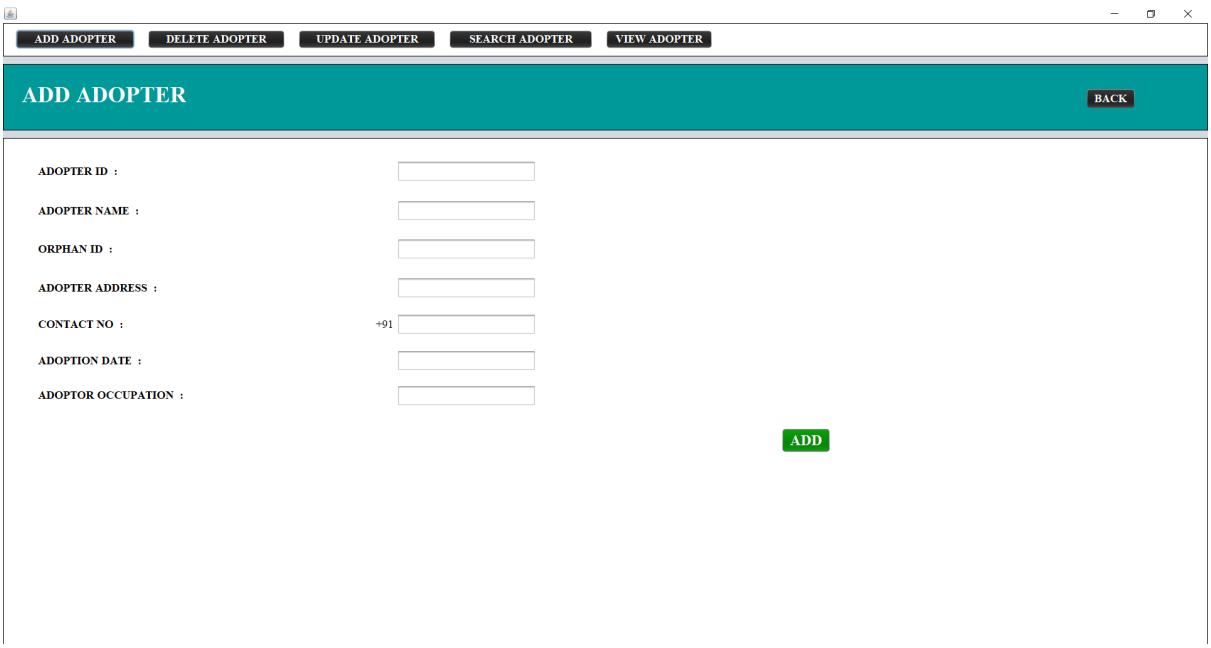

FIG 6.19: ADOPTER DETAILS

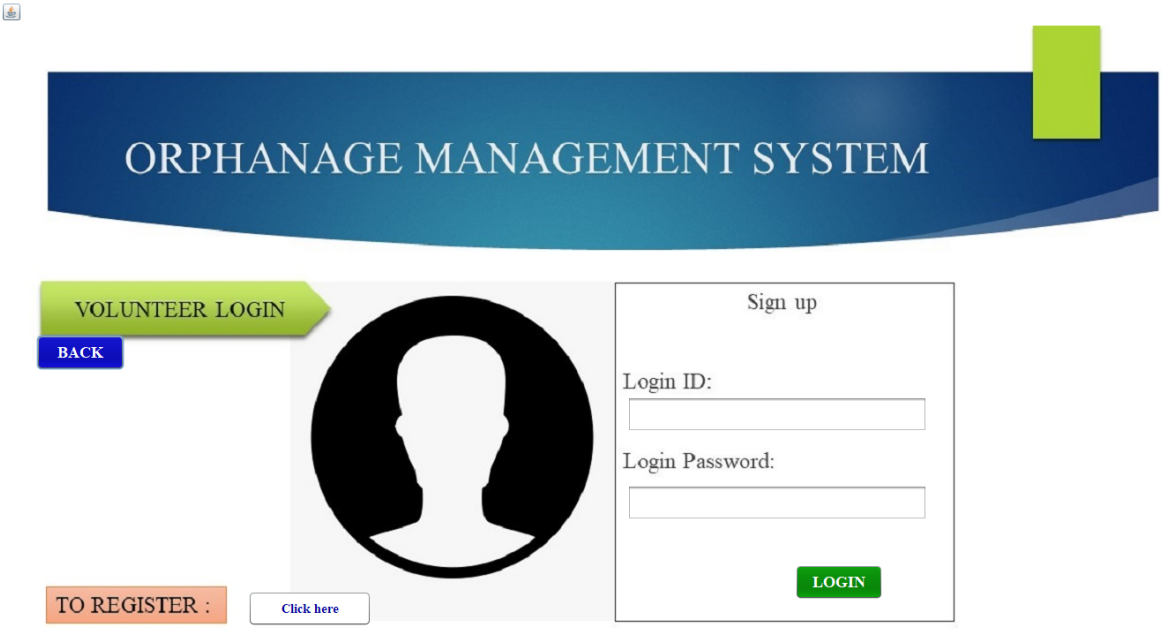

FIG 6.20: VOLUNTEER LOGIN PAGE

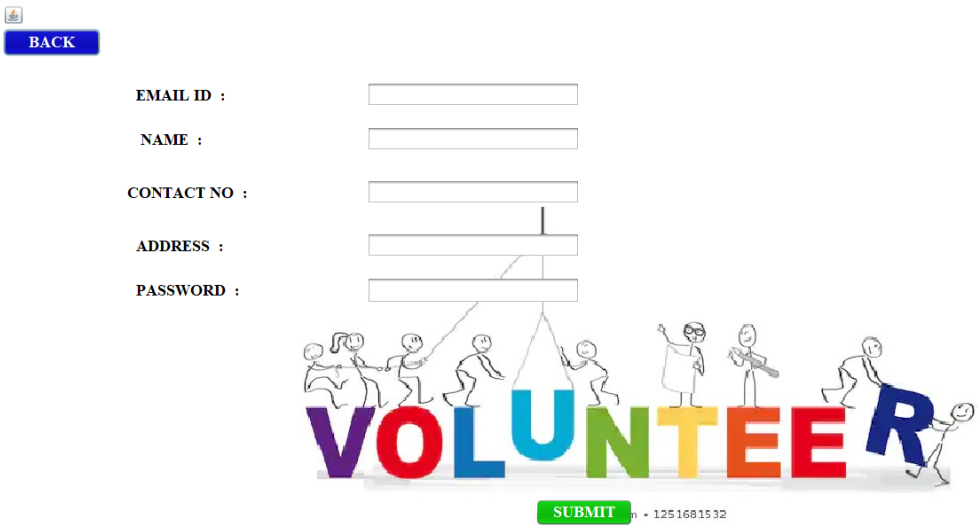

FIG 6.21: VOLUNTEER REGISTERATION PAGE

FIG 6.22: VOLUNTEERING PAGE

CHAPTER-7
CONCLUSION

CHAPTER 7

CONCLUSION

The orphanage project's primary objective is to meet the physical, mental, emotional and spiritual needs of orphaned and /or destitute children by providing them with love, care, nourishment, guidance, and the highest level of education so that they can grow to be values members of society with a promising future .
Here the admin remotely uses the software to handle the request of different users and organization.

Existing systems consists of only the orphanages details such as address contact numbers etc.it does not have any centralized and separate website for orphanages.

In existing system people can view only the address of the orphanages which are searched by the users. In that people cannot do any donation or sponsorships. This is the main drawback of the existing system. And also, this is the disadvantage of the orphanages those who are in need of getting help from others.

This online orphanage connecting system is developing to facilitate the orphanages. Here the user's visit the website by registering their details. Then they can view the orphanage details. Those who is willing to donate to the orphanage can donate by giving their details. Adoption option is also included with the eligibility checker.

The vision of the orphanage project is to one day have a self-sustaining community consisting of a children's home, orphanage and other facilities that help a community to succeed. Our vision is to one day establish such communities all over the world.

Benefits:

The project is identified by the merits of the system offered to the user. The merits of this project

are as follows: -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much

simplicity.

- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updating so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover, there is restriction for him that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.

CHAPTER-8
BIBLIOGRAPHY

CHAPTER 8

BIBLIOGRAPHY

REFERENCES

BOOK REFERENCE

[1] Database System Models, Languages, Ramez Elmasri and Sham Kant B. Navathe, 7th

WEBSITES

[1] www.stackoverflow.com

[2] www.youtube.com

[3] https://www.javatpoint.com/

[4] www.google.com

[5] https://docs.oracle.com/javase/ (framework)

ACTUAL CODE

ADMIN LOGIN:

```
String a=jTextField1.getText();
String b=jPasswordField1.getText();
 try{
        java.lang.Class.forName("com.mysql.jdbc.Driver");
         Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/orphan","root","nivyadhanu");
         String sql="select username,password from admin where username='"+jTextField1.getText()+"' and
password='"+jPasswordField1.getText()+"'";
          PreparedStatement prp = con.prepareStatement(sql);
         ResultSet rst=prp.executeQuery();
         int c=0;
         while(rst.next()){
            if(a.equalsIgnoreCase(rst.getString("username"))&& b.equals(rst.getString("password"))){
                c=c+1;
                if(c==1){
                 JOptionPane.showMessageDialog(null , "Login Successfull");
            jTextField1.setText(null);
            jPasswordField1.setText(null);
            AdminPortal ap=new AdminPortal();
```

```
            ap.setVisible(true);
            dispose();
}
            }
          }
           if(c!=1){
            JOptionPane.showMessageDialog(null , "INVALID  CREDENTIALS");
            jTextField1.setText(null);
            jPasswordField1.setText(null);
          }
        }
       catch(HeadlessException | ClassNotFoundException | SQLException e){
              JOptionPane.showMessageDialog(null ,"INVALID DATA"+e);
              }// TODO add your handling code here:
   }
```

ADD ORPHAN DETAILS:

```
try{
              java.lang. Class.forName("com.mysql.jdbc.Driver");
              Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/orphan","root","nivyadhanu");
              String sql = "insert into orphandetails values(?,?,?,?,?,?,?,?,?,?)";
              PreparedStatement prp = con.prepareStatement(sql);
              prp.setString(1, (jTextField1.getText()));
              prp.setString(2, (jTextField5.getText()));
              prp.setInt(3, Integer.parseInt(jTextField2.getText()));
              prp.setString(4, (jComboBox4.getSelectedItem().toString()));
                SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault());
                String d=sdf.format(jDateChooser1.getDate());
              prp.setString(5, d);
              prp.setString(6, (jComboBox2.getSelectedItem().toString()));
              prp.setString(7, (jComboBox1.getSelectedItem().toString()));
              prp.setString(8, (jTextField6.getText()));
              prp.setString(9, (jComboBox3.getSelectedItem().toString()));
              prp.setString(10,(jComboBox5.getSelectedItem().toString()));

              prp.executeUpdate();
              JOptionPane.showMessageDialog(null, "DATA SAVED");
              orphanadd sp=new orphanadd();
```

```
            sp.setVisible(true);
            con.close();
            dispose();

        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null , e);
        }       // TODO add your handling code here:
    }
```

DELETE ORPHAN DETAILS:

```
try{
        java.lang.Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Orphan","root","nivyadhanu");

        String sql="delete from orphandetails where orphan_id=?";
        PreparedStatement pstmt=conn.prepareStatement(sql);
        pstmt.setString(1,(jTextField1.getText()));
        pstmt.executeUpdate();
        JOptionPane.showMessageDialog(null, "RECORD DELETED SUCCESSFULLY");
        conn.close();
    }
        // TODO add your handling code here:
    catch(Exception e)
        {
            JOptionPane.showMessageDialog(null , e);
        }
    }
```

SEARCH ORPHAN DETAILS:

```
try{
        java.lang.Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Orphan","root","nivyadhanu");
        String sql="select * from orphandetails where orphan_id=?";
        PreparedStatement pstmt=conn.prepareStatement(sql);
```

```
    pstmt.setString(1,(jTextField1.getText()));

    ResultSet rs=pstmt.executeQuery();
     if(rs.next())
       {
          jTextField5.setText(rs.getString("orphan_name"));
          jTextField2.setText(rs.getString("orphan_age"));
           String gender=rs.getString("gender");
          jComboBox4.setSelectedItem(gender);
          jTextField4.setText(rs.getString("date_of_join"));
          String health=rs.getString("health");
          jComboBox2.setSelectedItem(health);
          String edu=rs.getString("education");
          jComboBox1.setSelectedItem(edu);
          jTextField6.setText(rs.getString("background_name"));
          String bat=rs.getString("batch_id");
          jComboBox3.setSelectedItem(bat);
           String adopt=rs.getString("adopt_or_not");
          jComboBox5.setSelectedItem(adopt);

       }
    else
       {
          JOptionPane.showMessageDialog(null, "RECORD NOT FOUND");
       }
  }
       catch(Exception e)
        {
           JOptionPane.showMessageDialog(null , e);
        }       // TODO add your handling code here:
}
```

UPDATE ORPHAN DETAILS:

```
try{

          java.lang.Class.forName("com.mysql.jdbc.Driver");
```

```
Connection conn=
DriverManager.getConnection("jdbc:mysql://localhost:3306/Orphan","root","nivyadhanu");

String sql="update orphandetails set orphan_name=?, orphan_age=?, gender=?, date_of_join=?, health=?, education=?, background_name=?, batch_id=?, adopt_or_not=? where orphan_id=?";

PreparedStatement prp=conn.prepareStatement(sql);
prp.setString(1, (jTextField5.getText()));
prp.setInt(2, Integer.parseInt(jTextField2.getText()));
prp.setString(3, (jComboBox4.getSelectedItem().toString()));
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault());
String d=sdf.format(jDateChooser1.getDate());
prp.setString(4, d);
prp.setString(5, (jComboBox2.getSelectedItem().toString()));
prp.setString(6, (jComboBox1.getSelectedItem().toString()));
prp.setString(7, (jTextField6.getText()));
prp.setString(8, (jComboBox3.getSelectedItem().toString()));
prp.setString(9,(jComboBox5.getSelectedItem().toString()));
prp.setString(10, (jTextField1.getText()));

prp.executeUpdate();
JOptionPane.showMessageDialog(null , "RECORD UPDATED SUCCESSFULLY");
```

```
        conn.close();

    }

        catch(Exception e)

        {

            JOptionPane.showMessageDialog(null , e);

        } // TODO add your handling code here:

  }
```

VIEW ORPHAN DETAILS:

```
private void DisplayTable()

  {

    try{

        java.lang.Class.forName("com.mysql.jdbc.Driver");

        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Orphan","root","nivyadhanu");

        String sql="select * from orphandetails";

        PreparedStatement prp=conn.prepareStatement(sql);

    ResultSet rs=prp.executeQuery();

    t1.setModel(DbUtils.resultSetToTableModel(rs));
```

```
    }

        catch(Exception e)

        {

            JOptionPane.showMessageDialog(null , e);

        }

  }
```

In the same way we will code for all the modules.